> a memory interface device interposed between the host applications and the memory device and operably coupled to receive memory access requests from the number of host applications, interact with the memory device on behalf of the number of host applications for servicing the memory access requests in accordance with the interface requirements for the memory device, and provide result/status information to the host applications in accordance with the interface requirements for each of the number of host applications."

Thus, the Applicant's claimed invention provides a memory interface that acts on behalf of host applications and provides result and status information back to the host applications. The host applications and memory device may have different interface requirements. For example, a host application interface may have a different bus width than the memory device interface. Or, another host application interface may have different interface cycle times than the memory device interface. The memory interface essentially converts or translates between the interface requirements of the host applications and the interface requirements of the memory device.

In contrast, Bauman discloses a caching system employing an address conflict detection scheme. In Bauman, processors directly forward memory requests to a storage controller. There is no indication that one or more of the processor interfaces are different from the storage controller interface. In fact, at column 7 lines 62 – 66, Bauman states: "Lines 30, 32, 34, and 36 represent IP/SC interfaces, which are each a collection of function, data, and address lines between the respective IP and the SC0 28. Lines 38, 40, 42, and 44 represent similar IP/SC interfaces between SC1 29 and IP4 39, IP5 41, IP6 43, and IP7 45, respectively." This suggests that all the interfaces of Bauman are the same.

Thus, Bauman fails to teach or suggest the Applicant's claimed invention, wherein a memory interface device is interposed between host applications and a memory device and operably coupled to receive memory access requests from the number of host applications, interact with the memory device on behalf of the number of host applications for servicing the memory access requests in accordance with the interface requirements for the memory device, and provide result/status information to the host applications in accordance with the interface requirements for each of the number of host applications. The Applicant therefore respectfully asserts that independent claim 32 is in condition for allowance.

Claim Rejections – 35 U.S.C. § 103

1. Claims 1, 3-12, 14, 17, 19-31, 33, and 35-47 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Bauman in view of Hughes (US patent no. 5,784,582). This rejection is respectfully traversed.

Applicant's independent claim 1 recites:

> "A memory interface device for interfacing a number of host applications to a memory device, the memory interface device comprising:
>
> a host interface for interfacing with the number of host applications;
>
> a memory interface for interfacing with the memory device wherein one or more of the host applications and the memory device have different interface requirements;
>
> a number of contexts operably coupled to the host interface for receiving memory access requests from the number of host applications and receiving memory access requests from the number of host applications and providing result/status information to the number of host applications; and
>
> control logic operably coupled to obtain memory access requests from the number of contexts, interact with the memory device over the memory interface for servicing the memory access requests on behalf of the number of host applications in accordance with the interface requirements for the memory device, and provide the result/status information to the number of host applications via the number of contexts in accordance with the interface requirements for each of the number of host applications."

As was previously described with regard to claim 32, Bauman fails to teach or suggest "control logic operably coupled to obtain memory access requests from the number of contexts, interact with the memory device over the memory interface for servicing the memory access requests on behalf of the number of host applications in accordance with the interface requirements for the memory device, and provide the result/status information to the number of host applications via the number of contexts in accordance with the interface requirements for each of the number of host applications". Any combination of Hughes with Bauman presents the same failing. Hughes adds nothing further to Bauman that would teach or suggest the claimed control logic. Rather, Hughes teaches that, but for the number of words transferred for a given operation, the interfaces to the SDRAM are the same (see Fig. 3 of Hughes). Therefore, both Bauman and Hughes, taken either alone or in combination, fail to teach or suggest the Applicant's claimed invention including the control logic as set forth in claim 1. The Applicant therefore respectfully asserts that claim 1, and its dependent claims 2 – 16, are in condition for allowance.

The Applicant's independent claim 17 recites program logic which includes control logic as is similarly recited in claim 1. The Applicant therefore respectfully asserts that claim 17 and its dependent claims 18 – 31 are in condition for allowance, for the same reasons as set forth with regard to claim 1.

2. Claims 2, 16, 18, 34, and 48 were rejected under 35 U.S.C. 103(a) as being unpatentable over Bauman and Hughes and further in view of Wentka (U.S. patent no. 5,968,114). This rejection is respectfully traversed.

Claims 2 and 16 depend from claim 1. Claim 18 depends from claim 17. Claims 34 and 48 depend from claim 32. As previously described, The combination of Bauman and Hughes fails to teach or suggest the Applicant's claimed control logic as set forth in claim 1 and 17 and memory interface as set forth in claim 32. Wentka adds nothing further to Bauman and Hughes to correct the previously set forth failings of the combination. The Applicant therefore respectfully asserts that Claims 2, 16, 18, 34, and 48 are in condition for allowance.

Conclusion

Accordingly, Applicant asserts that all the pending claims 1 – 48 are in condition for allowance. An indication of such is respectfully requested. Should further questions arise concerning this application, the Examiner is invited to call Applicant's attorney at the number listed below.

Respectfully Submitted,

Lindsay G Mc G

Lindsay G. McGuinness, Reg. No. 38,549
Mary M. Steubing, Reg. No. 37,946
Attorney for Applicant
Steubing McGuinness and Manaras LLP
Acton, MA 01720
(978) 264-6664

July 1, 2003

CLAIMS

1. (amended) A memory interface device for interfacing a number of host applications to a memory device, the memory interface device comprising:

a host interface for interfacing with the number of host applications;

a memory interface for interfacing with the memory device wherein one or more of the host applications and the memory device have different interface requirements;

a number of contexts operably coupled to the host interface for receiving memory access requests from the number of host applications and receiving memory access requests from the number of host applications and providing result/status information to the number of host applications; and

control logic operably coupled to obtain memory access requests from the number of contexts, interact with the memory device over the memory interface for servicing the memory access requests on behalf of the number of host applications in accordance with the interface requirements for the memory device, and provide the result/status information to the number of host applications via the number of contexts in accordance with the interface requirements for each of the number of host applications.

2. (original) The memory interface device of claim 1, wherein the number of host applications comprises a number of packet processing contexts of a packet processor, and wherein the host interface conforms to a packet processor interface.

3. (original) The memory interface device of claim 1, wherein the memory device comprises a content-addressable memory (CAM), and wherein the memory interface conforms to a CAM interface.

4. (original) The memory interface device of claim 1, wherein the number of contexts comprise a number of context registers sets.

5. (original) The memory interface device of claim 4, wherein each context register set corresponds to one and only one of the number of host applications.

6. (original) The memory interface device of claim 1, wherein the control logic comprises:

monitoring logic;

scheduling logic;

memory interface logic; and

result/status logic, wherein:

the monitoring logic is operably coupled to monitor the number of contexts for detecting memory access requests and providing the memory access requests to the scheduling logic;

the scheduling logic is operably coupled to schedule memory access operations for the memory access requests;

the memory interface logic is operably coupled to generate memory interface signals for interfacing with the memory device over the memory interface; and

the result/status logic is operably coupled to provide result/status information to the number of host application(s).

7. (original) The memory interface device of claim 6, wherein each context comprises a context register set, and wherein the monitoring logic is operably coupled to monitor a predetermined register in each context register set to detect a memory access request.

8. (original) The memory interface device of claim 7, wherein the predetermined register comprises an instruction register.

9. (original) The memory interface device of claim 6, wherein the memory interface supports pipelining of memory access operations, and wherein the scheduling logic is operably coupled to pipeline a plurality of memory access requests over the memory interface.

10. (original) The memory interface device of claim 9, wherein the scheduling logic is operably coupled to determine that a plurality of memory access request conflict and execute at least one of the conflicting memory access requests as an atomic operation.

11. (original) The memory interface device of claim 10, wherein the scheduling logic is operably coupled to clear the pipeline in order to execute the conflicting memory access request as an atomic operation.

12. (original) The memory interface device of claim 6, wherein the result/status logic is operably coupled to correlate result/status information information with its corresponding memory access request.

13. (original) The memory interface device of claim 6, wherein the result/status logic is operably coupled to store the result/status information for each memory access request in a corresponding context.

14. (original) The memory interface device of claim 13, wherein each context comprises a validity indicator, and wherein the result/status logic is operably coupled to set the validity indicator in each context when the corresponding memory access is complete and the result/status information is available.

15. (original) The memory interface device of claim 1 embodied as programmed programmable logic device.

16. (original) The memory interface device of claim 1 embodied as an application specific integrated circuit.

17. (amended) Program logic for programming a programmable logic device, the program logic comprising:

host interface logic for interfacing with [the] a number of host applications;

memory interface logic for interfacing with [the] a memory device wherein one or more of the host applications and the memory device have different interface requirements;

a number of contexts operably coupled to the host interface logic for receiving memory access requests from the number of host applications and providing result/status information to the number of host applications; and

control logic operably coupled to obtain memory access requests from the number of contexts, interact with the memory device using the memory interface logic for servicing the memory access requests on behalf of the number of host applications in accordance with the interface requirements for the memory device, and provide the result/status information to the number of host applications via the number of contexts in accordance with the interface requirements for each of the number of host applications.

18. (original) The program logic of claim 17, wherein the number of host applications comprises a number of packet processing contexts of a packet processor, and wherein the host interface logic conforms to a packet processor interface.

19. (original) The program logic of claim 17, wherein the memory device comprises a content-addressable memory (CAM), and wherein the memory interface logic conforms to a CAM interface.

20. (original) The program logic of claim 17, wherein the number of contexts comprises a number of context registers sets.

21. (original) The program logic of claim 20, wherein each context register set corresponds to one and only one of the number of host applications.

22. (original) The program logic of claim 17, wherein the control logic comprises:
monitoring logic;
scheduling logic;
memory interface logic; and
result/status logic, wherein:

the monitoring logic is operably coupled to monitor the number of contexts for detecting memory access requests and providing the memory access requests to the scheduling logic;

the scheduling logic is operably coupled to schedule memory access operations for the memory access requests;

the memory interface logic is operably coupled to generate memory interface signals for interfacing with the memory device using the memory interface logic; and

the result/status logic is operably coupled to provide result/status information to the number of host application(s).

23. (original) The program logic of claim 22, wherein each context comprises a context register set, and wherein the monitoring logic is operably coupled to monitor a predetermined register in each context register set to detect a memory access request.

24. (original) The program logic of claim 23, wherein the predetermined register comprises an instruction register.

25. (original) The program logic of claim 22, wherein the memory interface supports pipelining of memory access operations, and wherein the scheduling logic is operably coupled to pipeline a plurality of memory access requests over the memory interface.

26. (original) The program logic of claim 25, wherein the scheduling logic is operably coupled to determine that a plurality of memory access requests conflict and execute at least one of the conflicting memory access requests as an atomic operation.

27. (original) The program logic of claim 26, wherein the scheduling logic is operably coupled to clear the pipeline in order to execute the conflicting memory access request as an atomic operation.

28. (original) The program logic of claim 22, wherein the result/status logic is operably coupled to correlate result/status information with its corresponding memory access request.

29. (original) The program logic of claim 22, wherein the result/status logic is operably coupled to store the result/status information for each memory access request in a corresponding context.

30. (original) The program logic of claim 29, wherein each context comprises a validity indicator, and wherein the result/status logic is operably coupled to set the validity indicator in each context when the corresponding memory access is complete and the result/status information is available.

31. (original) The program logic of claim 17 embodied in a computer readable medium.

32. (amended) An apparatus comprising:

a number of host applications;

a memory device, wherein one or more of the host applications and the memory device have different interface requirements; and

a memory interface device interposed between the host applications and the memory device and operably coupled to receive memory access requests from the number of host applications, interact with the memory device on behalf of the number of host applications for servicing the memory access requests in accordance with the interface requirements for the memory device, and provide result/status information to the host applications in accordance with the interface requirements for each of the number of host applications.

33. (original) The apparatus of claim 32, wherein the memory interface device comprises:
a host interface for interfacing with the number of host applications;

a memory interface for interfacing with the memory device;

a number of contexts operably coupled to the host interface for receiving memory access requests from the number of host applications providing result/status information to the number of host applications; and

control logic operably coupled to obtain memory access requests from the number of contexts, interact with the memory device over the memory interface for servicing the memory access requests on behalf of the number of host applications, and provide the result/status information to the number of host applications via the number of contexts.

34. (original) The apparatus of claim 33, wherein the number of host applications comprises a number of packet processing contexts of a packet processor contexts of a packet processor, and wherein the host interface conforms to a packet processor interface.

35. (original) The apparatus of claim 33, wherein the memory device comprises a content-addressable memory (CAM), and wherein the memory interface conforms to a CAM interface.

36. (original) The apparatus of claim 33, wherein the number of contexts comprises a number of context registers sets.

37. (original) The apparatus of claim 36, wherein each context register set corresponds to one and only one of the number of host applications.

38. (original) The apparatus of claim 33, wherein the control logic comprises:
monitoring logic;
scheduling logic;
memory interface logic; and
result/status logic, wherein:

the monitoring logic is operably coupled to monitor the number of contexts for detecting memory access requests and providing the memory access requests to the scheduling logic;

the scheduling logic is operably coupled to schedule memory access operations for the memory access requests;

the memory interface logic is operably coupled to generate memory interface signals for interfacing with the memory device over the memory interface; and

the result/status logic is operably coupled to provide result/status information to the number of host application(s).

39. (original) The apparatus of claim 38, wherein each context comprises a context register set, and wherein the monitoring logic is operably coupled to monitor a predetermined register in each context register set to detect a memory access request.

40. (original) The apparatus of claim 39, wherein the predetermined register comprises an instruction register.

41. (original) The apparatus of claim 38, wherein the memory interface supports pipelining of memory access operations, and wherein the scheduling logic is operably coupled to a pipeline a plurality of memory access requests over the memory interface.

42. (original) The apparatus of claim 41, wherein the scheduling logic is operably coupled to determine that a plurality of memory access requests conflict and execute at least one of the conflicting memory access requests as an atomic operation.

43. (original) The apparatus of claim 42, wherein the scheduling logic is operably coupled to clear the pipeline in order to execute the conflicting memory access request as an atomic operation.

44. (original) The apparatus of claim 38, wherein the result/status logic is operably coupled to correlate result/status information with its corresponding memory access request.

45. (original) The apparatus of claim 38, wherein the result/status logic is operably coupled to store the result/status information for each memory access request in a corresponding context.

46. (original) The apparatus of claim 45, wherein each context comprises a validity indicator, and wherein the result/status logic is operably coupled to set the validity indicator in each context when the corresponding memory access is complete and the result/status information is available.

47. (original) The apparatus of claim 32, wherein the memory interface device is a programmed programmable logic device.

48. (original) The apparatus of claim 32, wherein the memory interface device is an application specific integrated circuit.